Secure Audit Service by using TPA

Pradnya P. Kulkarni
PG Student M B E Society's
College of Engineering,
Ambajogai, Maharashtra, India

ABSTRACT
Using Cloud Storage, users can remotely store their data and enjoy high quality applications and services from a shared pool of resources. However, users, when they don't have the physical possession of the data, chances of raising data integrity issue are possible. Thus, verification of data integrity is the at most important for a user, who has outsourced data to the cloud. To make the integrity check, a public auditing must be made possible. For it, resort to a Third Party Auditor (TPA). Also, the auditing process should not bring in further more burdens to the user, a secure cloud storage for which integrate the technique of Homomorphic linear authenticator with random masking. Thereby, assure integrity of the user's outsourced data in the cloud.

Keywords
Data integrity, Third Party Auditing, Public Auditing, Data storage, Cloud computing, Batch verification, Zero knowledge.

1. INTRODUCTION
Cloud storage such as Amazon, Yahoo, Google, Microsoft, and Mozy.com allows clients to store their data on remote storage. The data is stored remotely on remote storage and it can be accessed through the internet connection between client's machine and remote database on cloud. Storing data on cloud gives clients number of advantages like client don't have to maintain the data as it is maintained by the cloud service provider, pay only that they used, client can access his data from anywhere with the help of internet and he do not need to carry the physical data storage devices, enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources. Though these advantages make cloud storage a very economical option for storing data it has some drawbacks like the data loss incidents may take placed. Lots of internal and external threats are there and the data storage of client may be kept hidden from client to maintain reputation, there may be bugs in the network path or in the software. [1]. Cloud computing concerns some security like Confidentiality: unauthorized person cannot get any stored information. Integrity: Ensuring that information held in a system is a proper representation of the information intended and that it has not been modified by an unauthorized person. Availability: Ensuring that information processing resources are not made unavailable by malicious action. Non-Repudiation: Ensuring that agreements made electronically can be proven to have been made. As clients have limited capacity and they may do only uploading and downloading

Data from cloud storage. User downloads all data in order to check integrity of stored data it is very costly and tedious task. In the proposed system a Third Party Auditor (TPA) is introduced who will verify the data integrity of the client's data stored on cloud storage. TPA audit data when user needed. TPA has more potential than user and beneficial for cloud provider like audit result from TPA gives more values for Cloud base service platform and also they fulfill the cloud computing concerns. [2, 3]

Third party auditor (TPA), so met a) TPA audit cloud data storage without the local or original copy of data, and should not put any additional on-line burden to the cloud user; b) The third party auditing process should preserve user data privacy. To handle this problem use of homographic linear authentication (HLA) .By integrating HLA with random masking protocol guarantees that third party auditor could not learn anything about data content stored in cloud server during auditing processes

The aggregation and algebraic properties of the authenticator further benefits design for the batch auditing. Specifically, concentrate on the following aspects:

1. Enables an external auditor to audit user's cloud data without learning the data content.
2. Achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA in a privacy-preserving manner.
3. The security and justify the performance through concrete experiments and comparisons with the state of the art.

2. RELATED WORK
Ateniese et al. [9] are the first to consider public auditability in their "provable data possession" (PDP) model for ensuring possession of data files on untrusted storages. They utilize the RSA-based homomorphic linear authenticators for auditing outsourced data and suggest randomly sampling a few blocks of the file. However, among their two proposed schemes, the one with public auditability exposes the linear combination of sampled blocks to external auditor. When used directly, their protocol is not provably privacy preserving, and thus may leak user data information to the external auditor. Juels et al. [11] describe a "proof of retrievability" (PoR) model, where spot-checking and error -correcting codes are used to ensure both "possession" and "retrievability" of data files on remote archive service system.

However, the number of audit challenges a user can perform is fixed a prior, and public auditability is not supported in their main scheme. Although they describe a straightforward Markel-tree construction for public PoRs, this approach only works with encrypted data. Later, Bowers et al. [18] propose an improved framework for POR protocols that generalizes Juels' work. Dodis et al. [29] also give a study on different variants of PoR with private auditability. Shacham and Waters [13] design an improved PoR scheme built from BLS signatures [19] with proofs of security in the security

Model defined in [11]. Similar to the construction in [9], they use publicly verifiable homographic linear authenticators that are built from provably secure BLS signatures. Based on the elegant BLS construction, a compact and public verifiable

scheme is obtained. Again, their approach is not privacy preserving due to the same reason as [9]. Shah et al. [15], [10] propose introducing a TPA to keep online storage honest by first encrypting the data then sending a number of pre-computed symmetric-keyed hashes over the encrypted data to the auditor. The auditor verifies the integrity of the data file and the server's possession of a previously committed decryption key. This scheme only works for encrypted files, requires the auditor to maintain state, and suffers from bounded usage, which potentially brings in online burden to users when the keyed hashes are used up.

Dynamic data have also attracted attentions in the recent literature on efficiently providing the integrity guarantee of remotely stored data. Ateniese et al. [21] is the first to propose a partially dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. In [22], Wang et al. consider a similar support for partially dynamic data storage in a distributed scenario with additional feature of data error localization. In a subsequent work, Wang et al. [8] propose to combine BLS-based HLA with MHT to support fully data dynamics. Concurrently, Erway et al. [23] develop a skip list-based scheme to also enable provable data possession with full dynamics support. However, the verification in both protocols requires the linear combination of sampled blocks as an input, like the designs in [9], [13], and thus does not support privacy-preserving auditing.

In other related work, Sebe et al. [30] thoroughly study a set of requirements which ought to be satisfied for a remote data possession checking protocol to be of practical.

Their proposed protocol supports unlimited times of file integrity verifications and allows preset tradeoff between the protocol running time and the local storage burden at the user. Schwarz and Miller [31] propose the first study of checking the integrity of the remotely stored data across multiple distributed servers. Their approach is based on erasure-correcting code and efficient algebraic signatures, which also have the similar aggregation property as the homomorphic authenticator utilized. Curtmola et al. [32] aim to ensure data possession of multiple replicas across the distributed storage system. They extend the PDP scheme in [9] to cover multiple replicas without encoding each replica separately, providing guarantee that multiple copies of data are actually maintained. In [33], Bowers et al. utilize a two-layer erasure-correcting code structure on the remotely archived data and extend their POR model [18] to distributed scenario with high-data availability assurance. While all the above schemes provide methods for efficient auditing and provable assurance on the correctness of remotely stored data, almost none of them necessarily meet all the requirements for privacy-preserving public auditing of storage.

3. ARCHITECTURE

A cloud data storage service involving three different entities, as illustrated in the Figure: the cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources; the third-party auditor, who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to TPA while hoping to keep their data private from TPA. The data integrity threats toward users' data can come from both internal and external attacks at CS. Besides, CS can be self-interested. For their own benefits, such as to maintain reputation, CS might even decide to hide these data.

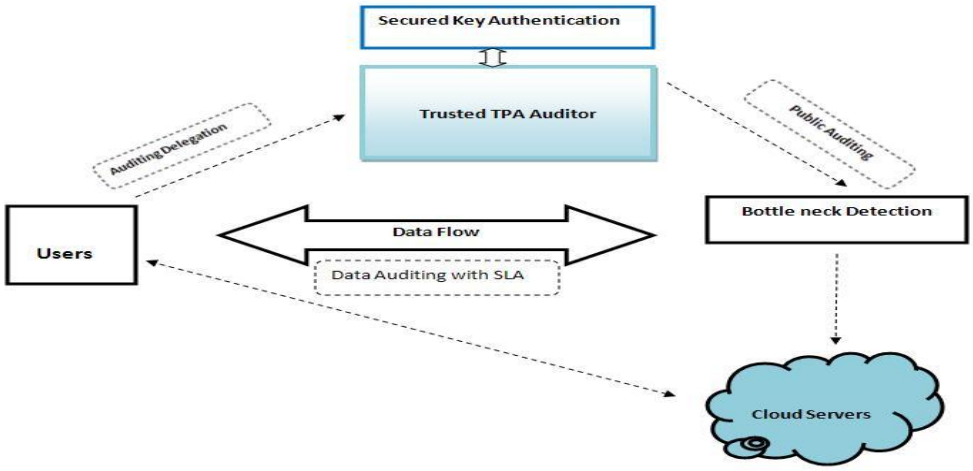

Fig 1: Secure Dynamic Trusted cloud Storage Architecture

The data integrity threats toward users' data can come from both internal and external attacks at CS. Besides, CS can be self-interested. For their own benefits, such as to maintain reputation, CS might even decide to hide these data corruption incidents to users. Using third-party auditing service provides a cost-effective method for users to gain trust in cloud. The TPA, who is in the business of auditing, is reliable and independent. However, it may harm the user if the TPA could learn the outsourced data after the audit. Note that in model, beyond users' reluctance to leak data to TPA, also assume that cloud servers have no incentives to reveal their hosted data to external parties. Therefore, neither CS nor TPA has motivations to collude with each other during the auditing process. To authorize the CS to respond to the audit delegated to TPA's the user can issue a certificate on authenticated against such a certificate. These authentication handshakes are

omitted in the following presentation. To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, protocol design should achieve the following security and performance guarantees:

a) Public auditability: To allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users.
b) Storage correctness: To ensure that there exists no cheating cloud server that can pass the TPA's audit without
c) Indeed storing users' data intact.
d) Privacy preserving: To ensure that the TPA cannot derive users' data content from the information collected during the auditing process.
e) Batch auditing: To enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.
f) Lightweight: To allow TPA to perform auditing with minimum communication and computation overhead.

3.1 Definitions and Framework

A similar definition of previously proposed schemes in the context of remote data integrity checking and adapt the framework for privacy-preserving public auditing system. A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, and VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of digital signatures. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA to audit the proof.
Proof of integrity setup into two steps:

- SETUP: -In set up phase user initializes public and secret Parameters of the system by executing Keygen algorithm and preprocess the data file F by using m_gen algorithm to generate the verification metadata. By deleting its local copy user will upload the data file on cloud server.
- AUDIT- In audit phase TPA send audit message or challenge to the cloud server to checking the stored data integrity. Random masking Homomorphic linear authenticator technique is used. Cloud server give the proof by recalled the data file F as it is and TPA will then

3.2 Zero Knowledge Public Auditing

Comparison on auditing time between batch and individual auditing, when α-fraction of 256 responses are invalid: Per task auditing time denotes the total auditing time divided by the number of tasks.

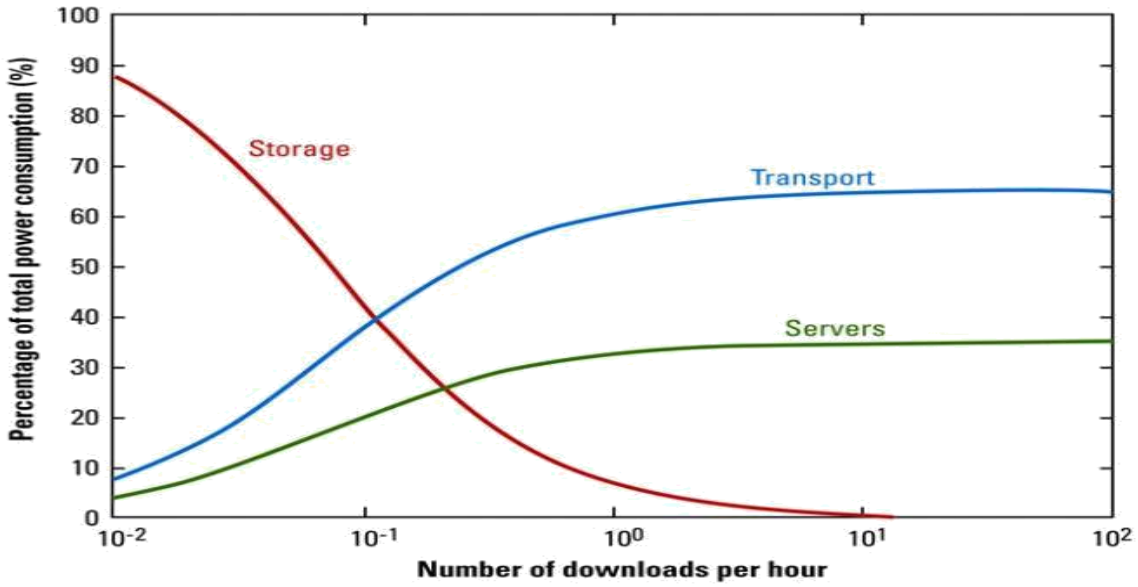

Fig 2: Performance comparison of Power Consumption.

The above auditing protocol achieves zero knowledge information leakage to the TPA, and it also ensures the storage correctness guarantee.

Proof:

Zero-knowledge is easy to see. Randomly pick ϒ; μ; & ζ

from Zp and ∑ from G1, set R<= e((Ⅱ $_{i=s1}^{sc}$ H(W $_i$)^{vi}) ^{ϒ} . u^{Ⅱ}, v) . e (g1, g) ^{ζ}/e (∑^{ϒ}; g) and back patch ϒ = h(R). For proof Of storage correctness, can extract ṕ similar to the extraction of μ'. Likewise, σ can be recovered from ∑. To conclude, a valid pair of σ and μ' can be extracted.

4. CONCLUSION

Secure cloud storage is achieved using two algorithms AES and HMAC which verifies the data integrity. This system not only reduces the load on client but also reduce fear of their outsourced data leakage. In this system TPA cannot audit only

one client at a time but also many clients simultaneously by using batch auditing scenario. The system is totally secure and highly efficient. The algorithm is partially homographic Encryption so using it, can be a future enhancement. Also a full fledge deployment of the application on public like handle large amount of data cloud can be an important future enhancement.TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users fear of their outsourced data leakage.

5. ACKNOWLEDGMENTS

I would like to thank my guide Dr. B. M. Patil for his help and guidance throughout this project and the semester, without him this would not have been possible.

6. REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010.

[2] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing," http://csrc.nist.gov/groups/SNS/cloud-computing/index.html, June 2009.

[3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009.

[4] Cloud Security Alliance, "Top threats to Cloud Computing [29].

[5] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions," http://www.techcrunch.com/2006/12/28/gmail-disasterreports-of-mass-email-deletions/, 2006.

[6] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," http:// www.techcrunch.com/2008/07/10/mediamaxthelinkup-closes-its-doors/, July 2008.

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Service Computing, vol. 5, no. 2, 220-232, Apr.-June 2012.

[8] F. Sebe, J. Domingo-Ferrer, A. Martı ´nez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient Remote Data Possession Checking in Critical Information Infrastructures," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 8, pp. 1034-1038, Aug. 2008.

[9] A. Juels and J. Burton, S. Kaliski, PORs: Proofs of Retrievability for Large Files," Proc. ACM Conf. Computer and Comm. Security (CCS'07), pp. 584-597, Oct. 2007.008.

[10] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, 2

[11] H. Shacham and B. Waters,"Compact Proofs of Retrievability," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances), in Cryptology (Asiacrypt vol. 5350, pp. 90-107, Dec. 2008.

[12] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan,"Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS '07), pp. 1-6, and 2007.

[13] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22,no. 5, pp. 847-859, May 2011.